

### **Remarks**

This RESPONSE is in reply to the Office Action mailed June 4, 2007. A Petition for Extension of Time is submitted herewith, together with the appropriate fee. No fee is due for the addition of new claims.

#### **I. Summary of Examiner's Rejections**

Prior to the Office Action mailed June 4, 2007, Claims 1-30 were pending in the Application. In the Office Action, the Oath/Declaration as originally filed was objected to as misspelling the inventor's name. The Drawings, Specification, and Claims 3, 6-30, 7, 17 and 27 were also objected to for minor informalities. Claims 1, 10, 11, 20, 21 and 30 were provisionally rejected on the grounds of non-statutory double patenting as being unpatentable over Claims 1, 9, 11, 19, 21 and 29 of co-pending Application No. 10/777,361. Claims 11-13 and 20 were rejected under 35 U.S.C. 102(e) as being anticipated by Kumar et al. (U.S. Patent No. 7,039,923, hereafter Kumar). Claims 1-10, 14-19 and 21-30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kumar in view of Susarla et al. (U.S. Patent No. 6,915, 511, hereafter Susarla).

#### **II. Summary of Applicant's Amendments**

The present Response provides a replacement Oath/Declaration, and amends the Drawings and the Specification. The present Response also cancels Claims 7-9, 17-19 and 27-29; amends Claims 1, 3, 6, 10-11, 13-14, 16, 20-21, 23-24, 26 and 30, and adds new Claims 31-39, leaving for the Examiner's present consideration Claims 1-6, 10-16, 20-26 and 30-39. Reconsideration of the Application, as amended, is respectfully requested.

#### **III. Oath/Declaration**

In the Office Action mailed June 4, 2007, the Oath/Declaration as originally filed was objected to as misspelling the inventor's name. Accordingly, submitted herewith is a corrected Oath/Declaration, showing the correct name of the inventor. Consideration thereof is respectfully requested.

**IV. Drawings**

In the Office Action mailed June 4, 2007, the Drawings were objected to for minor informalities. Accordingly, submitted herewith is a Replacement Sheet for Figure 1. Applicant respectfully submits that the proposed amendments to the Drawings are to correct various informalities, and that no new matter is being added.

**V. Specification**

In the Office Action mailed June 04, 2007, the Specification was objected to for minor informalities. Accordingly, the Specification has been amended as shown above. Applicant respectfully submits that the proposed amendments to the Specification are to correct various informalities, and that no new matter is being added.

**VI. Objections to the Claims**

In the Office Action mailed June 4, 2007, Claims 3, 6-30, 7, 17 and 27 were objected for various informalities. Accordingly, Claims 3, 6-30, 7, 17 and 27 have been amended as shown above to correct the informalities. Reconsideration thereof is respectfully requested.

**VII. Double Patenting Rejections**

In the Office Action mailed June 4, 2007, Claims 1, 10, 11, 20, 21 and 30 were provisionally rejected on the grounds of non-statutory double patenting as being unpatentable over Claims 1, 9, 11, 19, 21 and 29 of co-pending Application No. 10/777,361. Accordingly, Claims 1, 10, 11, 20, 21 and 30 have been amended as shown above, which Applicant believes renders the claims patentably distinct from Claims 1, 9, 11, 19, 21 and 29 of co-pending Application No. 10/777,361. Reconsideration thereof is respectfully requested.

**VIII. Claim Rejections under 35 U.S.C. §102**

In the Office Action mailed June 4, 2007, Claims 11-13 and 20 were rejected under 35 U.S.C. 102(e) as being anticipated by Kumar (U.S. Patent No. 7,039,923).

Claims 11-13 and 20 have been amended as shown above to more clearly define the embodiments therein. In particular, Claims 11-13 and 20 have been amended to define that the method therein comprises associating an application configuration file with a software application, wherein said configuration file includes a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server; parsing the configuration file, recognizing the modules and classes specified therein, and retrieving those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file; and wherein upon receiving a request to load the modules and classes of the software application, constructing an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader.

Applicant respectfully submits that this feature is neither disclosed by, nor obvious in view of the cited references. Reconsideration thereof is respectfully requested.

#### **IX. Claim Rejections under 35 U.S.C. §103**

In the Office Action mailed June 4, 2007, Claims 1-10, 14-19 and 21-30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kumar (U.S. Patent No. 7,039,923) in view of Susarla (U.S. Patent No. 6,915, 511).

#### **Claim 1**

Claim 1 has been amended to more clearly define the embodiment therein. As amended, Claim 1 defines:

1. *(Currently Amended) A system for loading software applications in an application server, comprising:  
an application server for executing a software application thereupon, wherein said software application has a plurality of modules and classes associated therewith;  
an application configuration file associated with said software application, wherein said configuration file includes a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server;*

*a deployment logic that parses the configuration file, recognizes the modules and classes specified therein, and retrieves those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file; and wherein upon receiving a request to load the modules and classes of the software application, the system constructs an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader.*

Claim 1 has been amended to more clearly define the embodiment therein as comprising an application server for executing a software application, wherein the software application has a plurality of modules and classes associated therewith. An application configuration file is used to define a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server. The configuration file is parsed to recognize the modules and classes specified therein, which are retrieved in a manner consistent with the tag layout. Upon receiving a request to load the modules and classes of the software application, the system constructs an application container at the application server with the classes and modules, in the hierarchical order in which the classes and modules were retrieved.

Kumar discloses a system and method for providing class dependency graph-based class loading and reloading that may be used to segregate namespaces in a graph-centric way, and may provide a set of normalized topologies that may be used to efficiently support hot-swapping of programmatic logic such as classes, applets, and beans, among other applications. Embodiments may provide a domain-independent, flexible and robust namespace segregation technique that is based on the dependency between the various classes and not on details like the roles the classes play. (Abstract).

Susarla discloses a system and method for providing dynamic class reloading using a modular, pluggable and maintainable class loader is described. Each application in an application server (or alternatively in any implementation) may include a dynamic class loader module. The class loader module may include a hierarchical stack of class loaders. Each module in the application may be associated with its own class loader. Each class loader may be responsible for loading one or more classes. When a class is changed, the changed class may be detected by the class loader module. (Abstract)

However, Applicant respectfully submits that, while the above-cited references disclose means for handling class dependencies, including the use of a hierarchical stack of class loaders, the references do not appear to disclose or suggest, either alone or in combination, the feature of a software application configuration file that uses a tag layout with application class-loader structure elements to determine the hierarchy of modules and classes of the application that are to be loaded in the application server. Nor do the references appear to disclose or suggest the feature of receiving a request to load the modules and classes of the software application, whereupon the system constructs an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader. Claim 1 has been amended to more clearly define these features.

In view of the above comments, Applicant respectfully submits that Claim 1 is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claim 21**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Claim 21 has been similarly amended by the current Response to more clearly define the embodiments therein. For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 21, as amended, is likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 2-10, 14-19 and 22-30**

Claims 7-9, 17-19 and 27-29 have been canceled by the present Response, rendering moot the rejection of these claims. Claims 2-6, 14-16, 22-26 and 30 depend from and include all of the features of one of Claims 1, 11, or 21. Claims 2-6, 14-16, 22-26 and 30 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the amendments to the independent claims, and the comments provided above. Reconsideration thereof is respectfully requested.

**X. Additional Amendments**

Claims 31-39 have been newly added by the present Response. Applicant respectfully requests that new Claims 31-39 be included in the Application, and considered therewith.

**XI. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. §1.136 for extending the time to respond up to and including December 4, 2007.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: December 4, 2007

By: /Karl F. Kenna/  
Karl F. Kenna  
Reg. No. 45,445

Customer No.: 23910  
FLIESLER MEYER LLP  
650 California Street, Fourteenth Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800